



expensr

Your budgeting buddy

Προδιαγραφές Λογισμικού και Υλοποίηση Συστήματος

Version 3.0

Γιάννης Καζαντζίδης - iakazant@ece.auth.gr
Κωνσταντίνος Μυλωνάς - kmylonas@ece.auth.gr
Νικήτας Καραγιώργος - nikitask@ece.auth.gr
Χαράλαμπος Κεμαλίδης - kemalidic@ece.auth.gr

21/06/2023

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
A. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers και του M. Smialek's.	0.1
A. Συμεωνίδης	29/3/2014	Μικρή αναθεώρηση – τροποποίηση ενοτήτων	0.1.3
X. Ζολώτας	10/4/2020	Μεγάλη αναθεώρηση – αφαίρεση ενοτήτων	0.4
X. Ζολώτας	15/4/2020	Μεγάλη αναθεώρηση – προσθήκη ενότητας REST προδιαγραφών	0.5.3
K. Παναγιώτου	25/4/2020	Μεγάλη αναθεώρηση – προσθήκη ενότητας Nodered περιγραφής	0.5.7
A. Συμεωνίδης	30/4/2020	Αναθεώρηση και τελική δομή προτύπου	0.6

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
A. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Γιάννης Καζαντζίδης	10	iakazant@ece.auth.gr
Κωνσταντίνος Μυλωνάς	10	kmylonas@ece.auth.gr
Νικήτας Καραγιώργος	10	nikitask@ece.auth.gr
Χαράλαμπος Κεμαλίδης	10	kemalidic@ece.auth.gr



Πίνακας περιεχομένων

Λίστα Σχημάτων	4
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	7
1.1 Πρότυπο Abstract factory.....	7
1.2 Πρότυπο Proxy.....	7
1.3 Πρότυπο Adapter	8
2. Αρχιτεκτονική Συστήματος.....	9
2.1 Αναγνώριση Πόρων (Resources) Συστήματος	9
2.2 Τεκμηρίωση REST διεπαφής	10
2.2.1 Πόρος PersonallInfo	10
2.2.2 Πόρος Income	12
2.2.3 Πόρος Expense	15
2.2.4 Πόρος Goal	18
2.2.5 Πόρος Notification	20
3. Υλοποίηση συστήματος με Node-RED	21
3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές Node-RED.....	21
3.1.1 Ροές πόρου PersonallInfo	21
3.1.2 Ροές πόρου Income.....	22
3.1.3 Ροές πόρου Expense	23
3.1.4 Ροές πόρου Goal	25
3.1.5 Ροές πόρου Notification.....	26
3.2 Υλοποίηση ιστοριών χρήστη	27
3.2.1 Ιστορία Χρήστη Add Expenses	27
3.2.2 Ιστορία Χρήστη Edit Expenses.....	27
3.2.3 Ιστορία Χρήστη Add Incomes.....	28
3.2.4 Ιστορία Χρήστη Edit Incomes.....	28
3.2.5 Ιστορία Χρήστη Manage Goals.....	29
3.2.6 Ιστορία Χρήστη Display Charts.....	29
3.2.7 Ιστορία Χρήστη Notify User for progress.....	30
3.2.8 Ιστορία Χρήστη Notify User for overconsumption	31
Παράρτημα Ι – Γλωσσάριο	32

Λίστα Σχημάτων

Εικόνα 1: Εφαρμογή προτύπου Abstract factory	7
Εικόνα 2: Εφαρμογή προτύπου Proxy	8
Εικόνα 3: Εφαρμογή προτύπου Adapter	8
Εικόνα 4: Μοντέλο δεδομένων User	10
Εικόνα 5: Παράμετροι του POST Endpoint του πόρου PersonalInfo.....	10
Εικόνα 6: Δοκιμή του POST Endpoint του πόρου PersonalInfo	10
Εικόνα 7: Αποκρίσεις του συστήματος για το POST Endpoint του πόρου PersonalInfo	10
Εικόνα 8: Παράμετροι του GET Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID	11
Εικόνα 9: Δοκιμή του GET Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID.....	11
Εικόνα 10: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID	11
Εικόνα 11: Παράμετροι του PUT Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID	11
Εικόνα 12: Δοκιμή του PUT Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID ...	11
Εικόνα 13: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID	11
Εικόνα 14: Μοντέλο δεδομένων Income	12
Εικόνα 15: Μοντέλο δεδομένων IncomesList, που αποτελείται από δεδομένα Income	12
Εικόνα 16: Παράμετροι του GET Endpoint του πόρου Income	12
Εικόνα 17: Δοκιμή του GET Endpoint του πόρου Income	12
Εικόνα 18: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Income	12
Εικόνα 19: Παράμετροι του POST Endpoint του πόρου Income	13
Εικόνα 20: Δοκιμή του POST Endpoint του πόρου Income	13
Εικόνα 21: Αποκρίσεις του συστήματος για το POST Endpoint του πόρου Income	13
Εικόνα 22: Παράμετροι του GET Endpoint του πόρου Income, για συγκεκριμένο Income ID	13
Εικόνα 23: Δοκιμή του GET Endpoint του πόρου Income, για συγκεκριμένο Income ID	13
Εικόνα 24: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Income, για συγκεκριμένο Income ID	13
Εικόνα 25: Παράμετροι του PUT Endpoint του πόρου Income, για συγκεκριμένο Income ID	14
Εικόνα 26: Δοκιμή του PUT Endpoint του πόρου Income, για συγκεκριμένο Income ID	14
Εικόνα 27: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου Income, για συγκεκριμένο Income ID	14
Εικόνα 28: Παράμετροι του DELETE Endpoint του πόρου Income, για συγκεκριμένο Income ID	14
Εικόνα 29: Δοκιμή του DELETE Endpoint του πόρου Income, για συγκεκριμένο Income ID..	14
Εικόνα 30: Αποκρίσεις του συστήματος για το DELETE Endpoint του πόρου Income, για συγκεκριμένο Income ID	14
Εικόνα 31: Μοντέλο δεδομένων Expense	15
Εικόνα 32: Μοντέλο δεδομένων ExpensesList, που αποτελείται από δεδομένα Expense.....	15
Εικόνα 33: Παράμετροι του GET Endpoint του πόρου Expense.....	15
Εικόνα 34: Δοκιμή του GET Endpoint του πόρου Expense	15
Εικόνα 35: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Expense	15
Εικόνα 36: Παράμετροι του POST Endpoint του πόρου Expense.....	16
Εικόνα 37: Δοκιμή του POST Endpoint του πόρου Expense	16

Εικόνα 38: Αποκρίσεις του συστήματος για το POST Endpoint του πόρου Expense.....	16
Εικόνα 39: Παράμετροι του GET Endpoint του πόρου Expense, για συγκεκριμένο Expense ID	16
Εικόνα 40: Δοκιμή του GET Endpoint του πόρου Expense, για συγκεκριμένο Expense ID.....	16
Εικόνα 41: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Expense, για συγκεκριμένο Expense ID.....	16
Εικόνα 42: Παράμετροι του PUT Endpoint του πόρου Expense, για συγκεκριμένο Expense ID	17
Εικόνα 43: Δοκιμή του PUT Endpoint του πόρου Expense, για συγκεκριμένο Expense ID.....	17
Εικόνα 44: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου Expense, για συγκεκριμένο Expense ID.....	17
Εικόνα 45: Παράμετροι του DELETE Endpoint του πόρου Expense, για συγκεκριμένο Expense ID	17
Εικόνα 46: Δοκιμή του DELETE Endpoint του πόρου Expense, για συγκεκριμένο Expense ID.....	17
Εικόνα 47: Αποκρίσεις του συστήματος για το DELETE Endpoint του πόρου Expense, για συγκεκριμένο Expense ID.....	17
Εικόνα 48: Μοντέλο δεδομένων Goal	18
Εικόνα 49: Μοντέλο δεδομένων GoalsList, που αποτελείται από δεδομένα Goal	18
Εικόνα 50: Παράμετροι του GET Endpoint του πόρου Goal.....	18
Εικόνα 51: Δοκιμή του GET Endpoint του πόρου Goal.....	18
Εικόνα 52: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Goal.....	18
Εικόνα 53: Παράμετροι του GET Endpoint του πόρου Goal, για συγκεκριμένο Goal ID	19
Εικόνα 54: Δοκιμή του GET Endpoint του πόρου Goal, για συγκεκριμένο Goal ID.....	19
Εικόνα 55: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Goal, για συγκεκριμένο Goal ID.....	19
Εικόνα 56: Παράμετροι του PUT Endpoint του πόρου Goal, για συγκεκριμένο Goal ID	19
Εικόνα 57: Δοκιμή του PUT Endpoint του πόρου Goal, για συγκεκριμένο Goal ID	19
Εικόνα 58: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου Goal, για συγκεκριμένο Goal ID.....	19
Εικόνα 59: Μοντέλο δεδομένων Notification.....	20
Εικόνα 60: Μοντέλο δεδομένων NotificationsList, που αποτελείται από δεδομένα Notification.....	20
Εικόνα 61: Παράμετροι του GET Endpoint του πόρου Notification	20
Εικόνα 62: Δοκιμή του GET Endpoint του πόρου Notification	20
Εικόνα 63: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Notification	20
Εικόνα 64: Ροή endpoint GET /user/{userID}/personalinfo	21
Εικόνα 65: Ροή endpoint POST /user/personalinfo	21
Εικόνα 66: Ροή endpoint PUT /user/{userID}/personalinfo	22
Εικόνα 67: Ροή endpoint GET /user/{userID}/income	22
Εικόνα 68: Ροή endpoint POST /user/{userID}/income	22
Εικόνα 69: Ροή endpoint GET /user/{userID}/income/{incomeID}	23
Εικόνα 70: Ροή endpoint PUT /user/{userID}/income/{incomeID}	23
Εικόνα 71: Ροή endpoint DELETE /user/{userID}/income/{incomeID}.....	23
Εικόνα 72: Ροή endpoint GET /user/{userID}/expense	23
Εικόνα 73: Ροή endpoint POST /user/{userID}/expense	24
Εικόνα 74: Ροή endpoint GET /user/{userID}/expense/{expenseID}	24
Εικόνα 75: Ροή endpoint PUT /user/{userID}/expense/{expenseID}	24

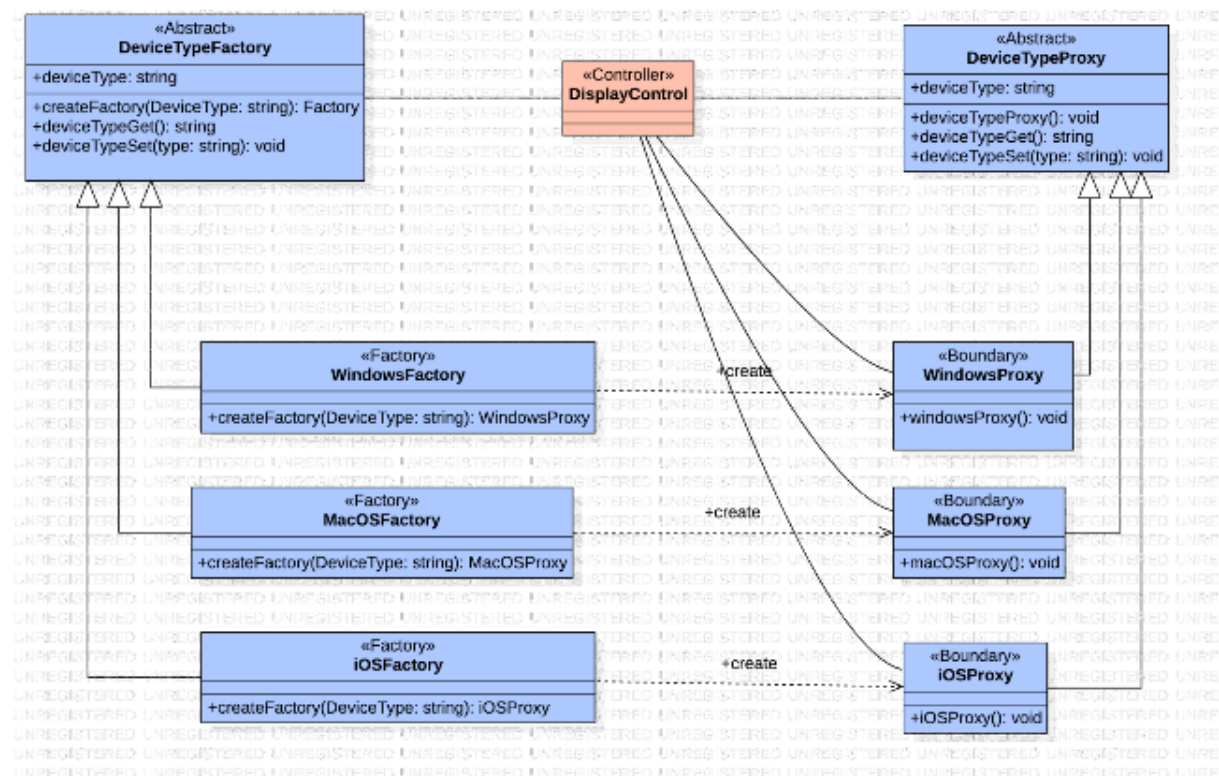


Εικόνα 76: Ποή endpoint DELETE /user/{userID}/expense/{expenseID}	25
Εικόνα 77: Ποή endpoint GET /user/{userID}/goal	25
Εικόνα 78: Ποή endpoint GET /user/{userID}/goal/{goalID}.....	25
Εικόνα 79: Ποή endpoint PUT /user/{userID}/goal/{goalID}.....	25
Εικόνα 80: Ποή endpoint GET /user/{userID}/notifications.....	26
Εικόνα 81: Ιστορία Χρήστη Add Expenses σε Node Red.....	27
Εικόνα 82: Ιστορία Χρήστη Edit Expenses σε Node Red	27
Εικόνα 83: Ιστορία Χρήστη Add Incomes σε Node Red	28
Εικόνα 84: Ιστορία Χρήστη Edit Incomes σε Node Red	28
Εικόνα 85: Ιστορία Χρήστη Manage Goals σε Node Red	29
Εικόνα 86: Ιστορία Χρήστη Display Charts σε Node Red	29
Εικόνα 87: Ιστορία Χρήστη Notify User for progress σε Node Red	30
Εικόνα 88: Ιστορία Χρήστη Notify User for overconsumption σε Node Red.....	31

1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

1.1 Πρότυπο Abstract factory

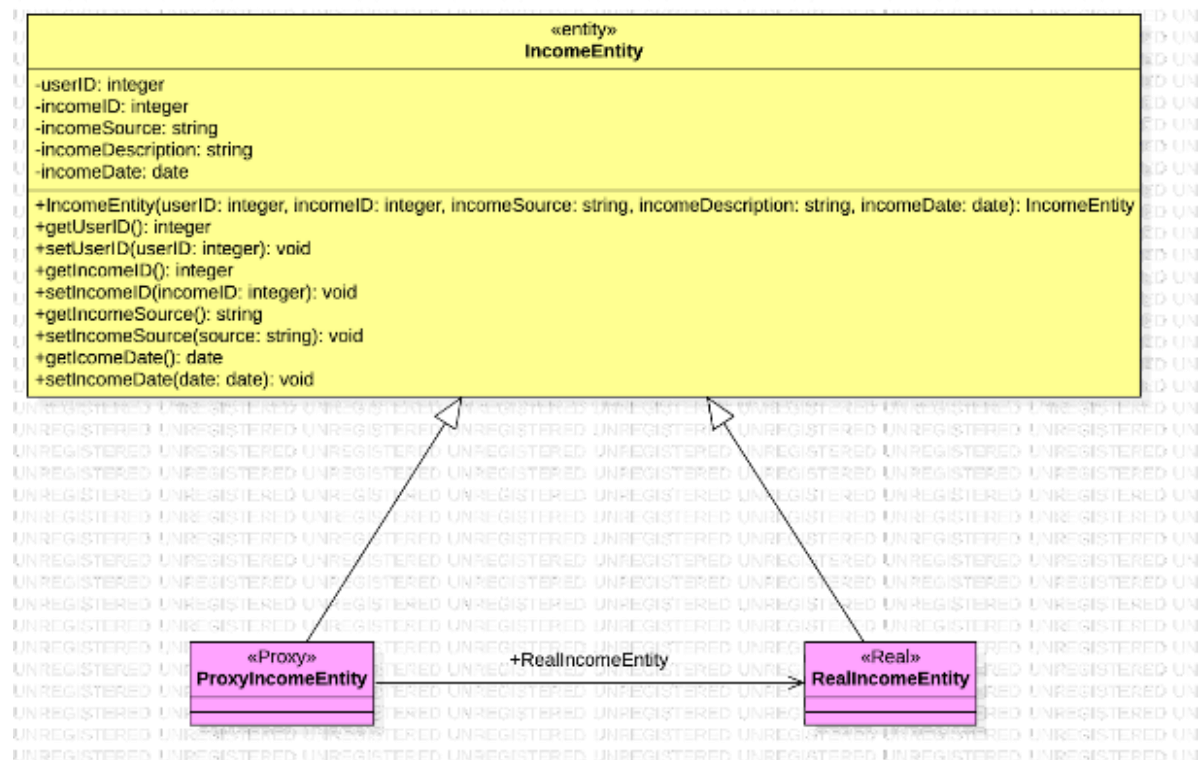
Το πρότυπο Abstract Factory είναι ένα **δημιουργικό πρότυπο**. Με αυτό το σχεδιαστικό πρότυπο, το σύστημά μας μπορεί να υποστηρίξει διαφορετικού τύπου συσκευές, Desktop mode και mobile mode (Windows, MacOS, iOS), ενώ καθιστά δυνατή και την επέκταση της εφαρμογής και σε άλλα λειτουργικά συστήματα. Έτσι, καλύπτεται η ΜΛΑ-1.



Εικόνα 1: Εφαρμογή προτύπου Abstract factory

1.2 Πρότυπο Proxy

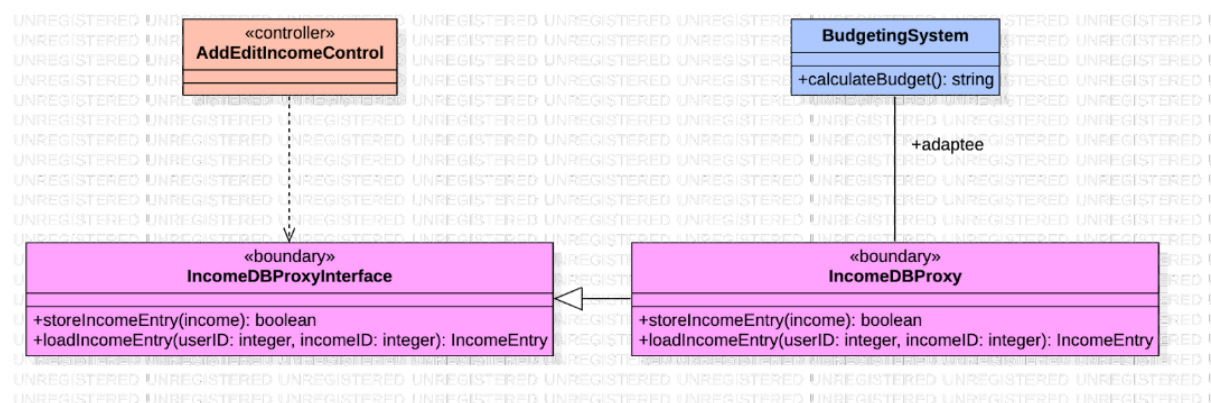
Το πρότυπο Proxy είναι ένα **δομικό πρότυπο**. Με αυτό το σχεδιαστικό πρότυπο δομείται το σύστημα μας ώστε να εξασφαλίζει την προστασία των προσωπικών δεδομένων των χρηστών σύμφωνα με τον γενικό κανονισμό GDPR. Έτσι, καλύπτεται η ΜΛΑ-2. Επίσης με αυτό το πρότυπο μπορούμε να δημιουργήσουμε place holders των οποίων η χρήση επιταχύνει τη λειτουργία του συστήματος. Με αυτό τον τρόπο καλύπτεται και η ΜΛΑ-4, που απαιτεί χρόνο απόκρισης 0.1 seconds.



Εικόνα 2: Εφαρμογή προτύπου Proxy

1.3 Πρότυπο Adapter

Το πρότυπο Adapter είναι ένα **δομικό πρότυπο**. Με αυτό το πρότυπο γίνεται δυνατή η συνεργασία του γραφικού περιβάλλοντος διεπαφής του χρήστη με διεπαφές του συστήματος. Έτσι καλύπτεται η ΜΛΑ-3.



Εικόνα 3: Εφαρμογή προτύπου Adapter

2. Αρχιτεκτονική Συστήματος

Χρησιμοποιήθηκε το εργαλείο SwaggerHub για την ακόλουθη ενότητα.

- [Σύνδεσμος](#) για το API στο SwaggerHub.
- [Σύνδεσμος](#) για το αρχείο τύπου JSON με τις προδιαγραφές του API.
- [Σύνδεσμος](#) για το αρχείο τύπου ZIP που περιέχει τον κώδικα για τον server της εφαρμογής

2.1 Αναγνώριση Πόρων (Resources) Συστήματος

Πίνακας 1: Πίνακας Κλάσεων BEC, Πόρων REST και Endpoints

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
PersonalInfo, (User)	/user/personalinfo	GET
PersonalInfo, (User)	/user/{userID}/personalinfo	POST, PUT
Income, (User)	/user/{userID}/income	GET, POST
Income, (User)	/user/{userID}/income/{incomeID}	GET, PUT, DELETE
Expense, (User)	/user/{userID}/expense	GET, POST
Expense, (User)	/user/{userID}/expense/{expenseID}	GET, PUT, DELETE
Goal, (User)	/user/{userID}/goal	GET
Goal, (User)	/user/{userID}/goal/{goalID}	GET, PUT
Notification, (User)	/user/{userID}/notifications	GET

2.2 Τεκμηρίωση REST διεπαφής

2.2.1 Πόρος PersonalInfo

2.2.1.1 Μοντέλο δεδομένων User

```
User {
  userID integer
  username string
  fullname string
  password string
  email string
  age integer
}
```

Εικόνα 4: Μοντέλο δεδομένων User

2.2.1.2 Endpoint POST πόρου PersonalInfo

POST

/user/personalinfo

Post personal info when creating account

⌵

Parameters

Cancel

No parameters

Request body required

application/json

User Schema

Examples:

An example of a user

```
{
  "userId": 1,
  "username": "admin",
  "fullName": "John Doe",
  "password": "password",
  "email": "john.doe@example.com",
  "age": 30
}
```

Εικόνα 5: Παράμετροι του POST Endpoint του πόρου PersonalInfo

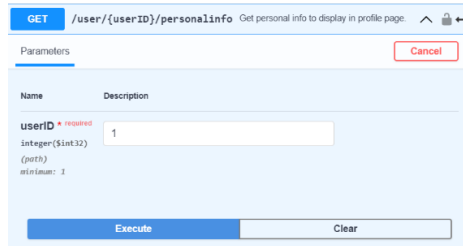
[illegible]

Εικόνα 6: Δοκιμή του POST Endpoint του πόρου PersonalInfo

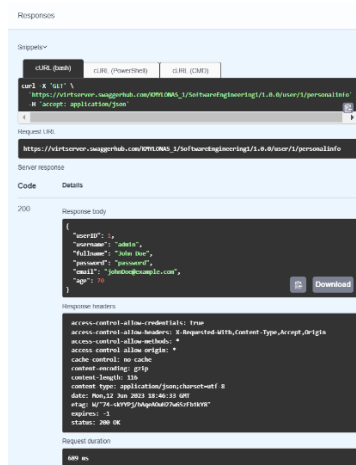
Code	Description	Links
401	Successful account creation	No links
	<p>Mode type Examples</p> <p>application/json An example of a user</p> <p>Content-type header</p> <p>Example Value Schema</p> <pre>{ "secret": " ", "username": "JohnDoe", "password": "JohnDoe", "email": "john.doe@company.com", "age": 22 }</pre>	
402	Bad Request	No links
	<p>Mode type Examples</p> <p>application/json An example of error</p> <p>Example Value Schema</p> <pre>{ "code": "BAD", "message": "Bad request!" }</pre>	
429	Conflict	No links
	<p>Mode type Examples</p> <p>application/json An example of error</p> <p>Example Value Schema</p> <pre>{ "code": "BAD", "message": "Conflict!" }</pre>	

Εικόνα 7: Αποκρίσεις του συστήματος για το POST Endpoint του πόρου PersonalInfo

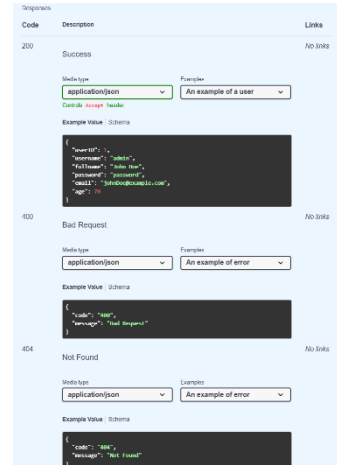
2.2.1.3 Endpoint GET πόρου PersonalInfo, για συγκεκριμένο User ID



Εικόνα 8: Παράμετροι του GET Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID

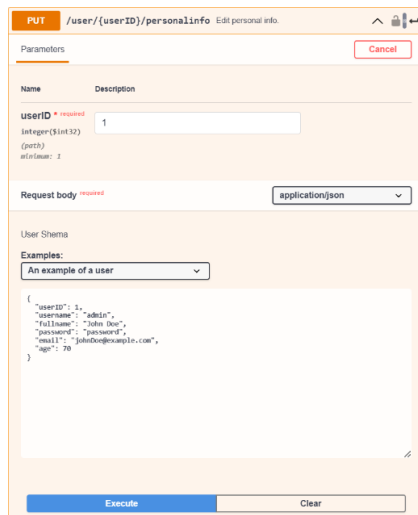


Εικόνα 9: Δοκιμή του GET Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID

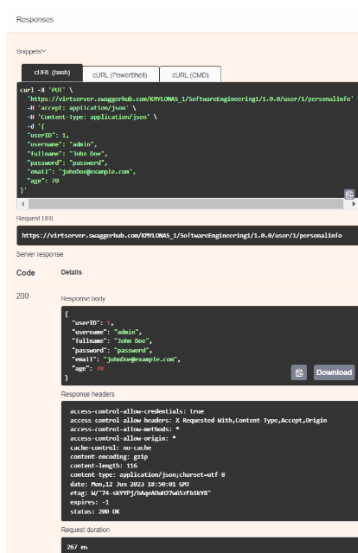


Εικόνα 10: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID

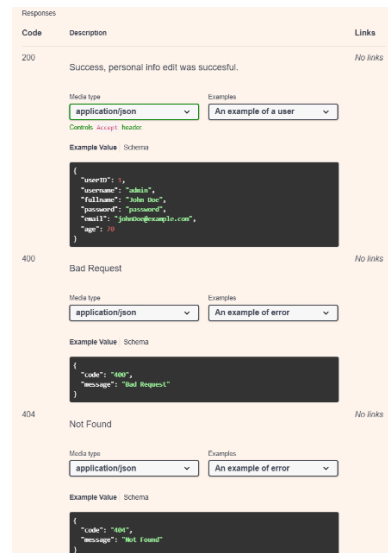
2.2.1.4 Endpoint PUT πόρου PersonalInfo, για συγκεκριμένο User ID



Εικόνα 11: Παράμετροι του PUT Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID



Εικόνα 12: Δοκιμή του PUT Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID



Εικόνα 13: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου PersonalInfo, για συγκεκριμένο User ID

2.2.2 Πόρος Income

2.2.2.1 Μοντέλο δεδομένων Income

```
Income {
  incomeID* integer
  userID* integer
  incomeAmount number($float)
  incomeSource string
  incomeDescription string
  incomeDate* string($date-time)
}
```

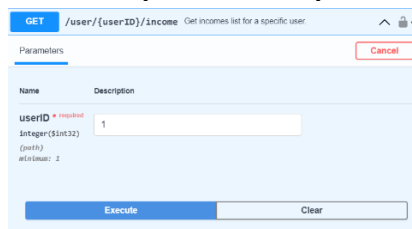
Εικόνα 14: Μοντέλο δεδομένων Income

2.2.2.2 Μοντέλο δεδομένων IncomesList

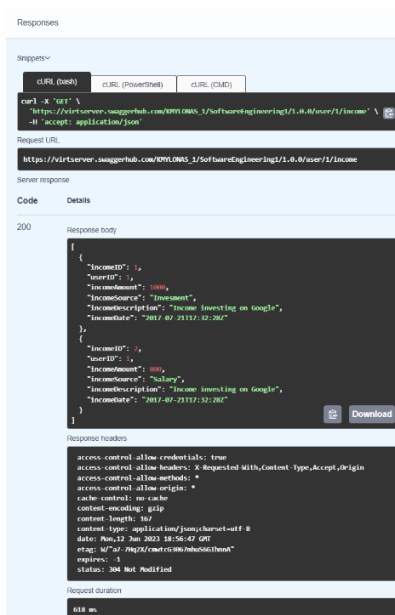
```
IncomesList [IncomesList {
  incomeID* integer
  userID* integer
  incomeAmount number($float)
  incomeSource string
  incomeDescription string
  incomeDate* string($date-time)
}]
```

Εικόνα 15: Μοντέλο δεδομένων IncomesList, που αποτελείται από δεδομένα Income

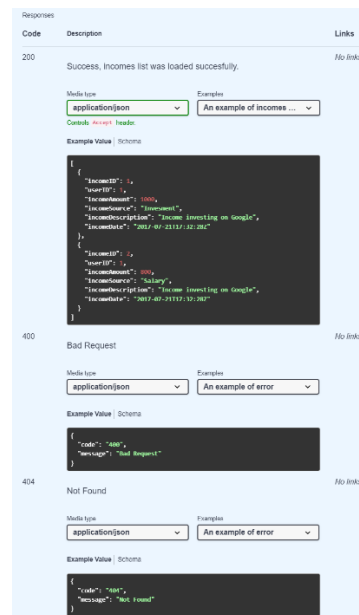
2.2.2.3 Endpoint GET πόρου Income



Εικόνα 16: Παράμετροι του GET Endpoint του πόρου Income

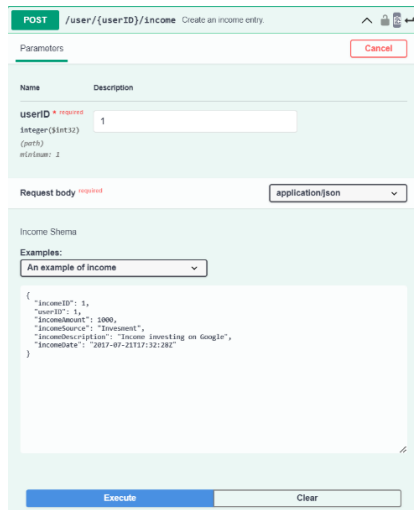


Εικόνα 17: Δοκιμή του GET Endpoint του πόρου Income



Εικόνα 18: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Income

2.2.2.4 Endpoint POST πόρου Income



POST /user/{userID}/income Create an income entry.

Parameters

Name	Description
userID *	required
integer(int32)	(path)
incomeID	optional
integer(int32)	(path)

Request body: application/json

Income Schema

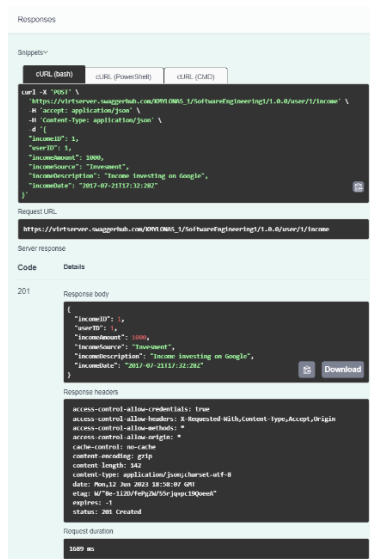
Examples:

An example of income

```
{
  "incomeID": 1,
  "userID": 1,
  "incomeAmount": 1000,
  "incomeSource": "Investment",
  "incomeDescription": "Income Investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}
```

Execute Clear

Εικόνα 19: Παράμετροι του POST Endpoint του πόρου Income



Responses

201 Successful income creation.

Media type: application/json

Example Value: An example of income

```
{
  "incomeID": 1,
  "userID": 1,
  "incomeAmount": 1000,
  "incomeSource": "Investment",
  "incomeDescription": "Income Investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}
```

Bad Request

Media type: application/json

Example Value: An example of error

```
{
  "code": "400",
  "message": "Bad Request"
}
```

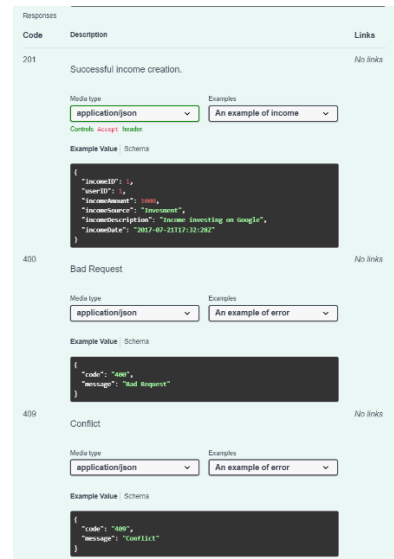
Conflict

Media type: application/json

Example Value: An example of error

```
{
  "code": "409",
  "message": "Conflict"
}
```

Εικόνα 20: Δοκιμή του POST Endpoint του πόρου Income



Responses

201 Successful income creation.

Media type: application/json

Example Value: An example of income

```
{
  "incomeID": 1,
  "userID": 1,
  "incomeAmount": 1000,
  "incomeSource": "Investment",
  "incomeDescription": "Income Investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}
```

Bad Request

Media type: application/json

Example Value: An example of error

```
{
  "code": "400",
  "message": "Bad Request"
}
```

Conflict

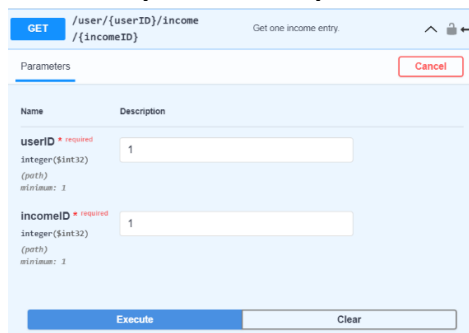
Media type: application/json

Example Value: An example of error

```
{
  "code": "409",
  "message": "Conflict"
}
```

Εικόνα 21: Αποκρίσεις του συστήματος για το POST Endpoint του πόρου Income

2.2.2.4 Endpoint GET πόρου Income, για συγκεκριμένο Income ID



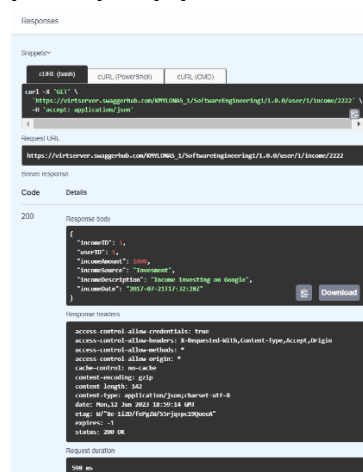
GET /user/{userID}/income/{incomeID} Get one income entry.

Parameters

Name	Description
userID *	required
integer(int32)	(path)
incomeID *	required
integer(int32)	(path)

Execute Clear

Εικόνα 22: Παράμετροι του GET Endpoint του πόρου Income, για συγκεκριμένο Income ID



Responses

200 Success, income was loaded successfully.

Media type: application/json

Example Value: An example of income

```
{
  "incomeID": 1,
  "userID": 1,
  "incomeAmount": 1000,
  "incomeSource": "Investment",
  "incomeDescription": "Income Investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}
```

Bad Request

Media type: application/json

Example Value: An example of error

```
{
  "code": "400",
  "message": "Bad Request"
}
```

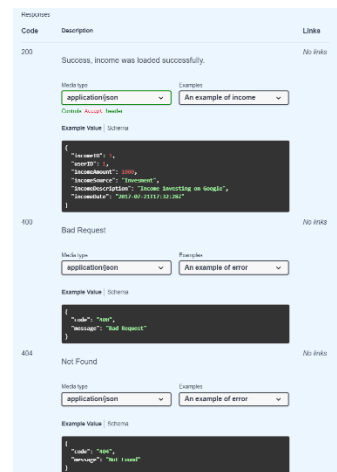
Not Found

Media type: application/json

Example Value: An example of error

```
{
  "code": "404",
  "message": "Not Found"
}
```

Εικόνα 23: Δοκιμή του GET Endpoint του πόρου Income, για συγκεκριμένο Income ID



Responses

200 Success, income was loaded successfully.

Media type: application/json

Example Value: An example of income

```
{
  "incomeID": 1,
  "userID": 1,
  "incomeAmount": 1000,
  "incomeSource": "Investment",
  "incomeDescription": "Income Investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}
```

Bad Request

Media type: application/json

Example Value: An example of error

```
{
  "code": "400",
  "message": "Bad Request"
}
```

Not Found

Media type: application/json

Example Value: An example of error

```
{
  "code": "404",
  "message": "Not Found"
}
```

Εικόνα 24: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Income, για συγκεκριμένο Income ID

PUT /user/{userID}/income
/{incomeID}

Edit one income entry

Parameters

Name
Description

userID ^{required}

Integer (32bit)

(path)

allowance: 1

IncomeID ^{required}

Integer (32bit)

(path)

allowance: 1

Request body ^{required}

application/json

Income Schema

Examples:

An example of Income

```

{
  "incomeID": 1,
  "userID": 1,
  "incomeSource": "Income",
  "incomeDescription": "Income investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}

```

Response

Code
Description
Links

200

Success, income was edited successfully.

Media type

application/json

Examples

Accept

An example of income

Example Value

Schema

```

{
  "incomeID": 1,
  "userID": 1,
  "incomeSource": "Income",
  "incomeDescription": "Income investing on Google",
  "incomeDate": "2017-07-21T17:32:28Z"
}

```

400

Bad Request

Media type

application/json

Examples

Accept

An example of error

Example Value

Schema

```

{
  "code": "BAD",
  "message": "Bad Request"
}

```

404

Not Found

Media type

application/json

Examples

Accept

An example of error

Example Value

Schema

```

{
  "code": "BAD",
  "message": "Not Found"
}

```

Εικόνα 27: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου Income, για συγκεκριμένο Income ID

Εικόνα 30: Αποκρίσεις του συστήματος για το DELETE Endpoint του πόρου Income, για συγκεκριμένο Income ID

2.2.3 Πόρος Expense

2.2.3.1 Μοντέλο δεδομένων Expense

```
Expense {  
  expenseID* integer  
  userID* integer  
  expenseAmount number($float)  
  expenseSource string  
  expenseDescription string  
  expenseDate* string($date-time)  
}
```

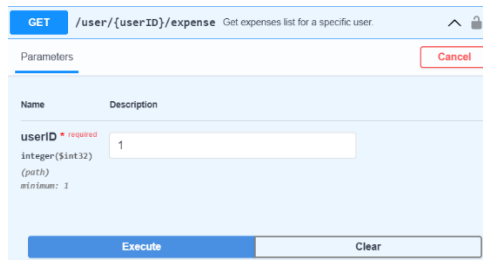
Εικόνα 31: Μοντέλο δεδομένων Expense

2.2.3.2 Μοντέλο δεδομένων ExpensesList

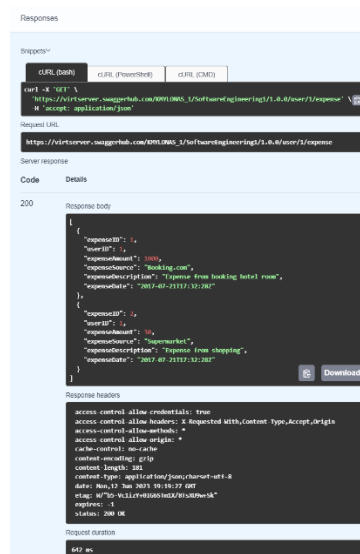
```
ExpensesList { ExpensesList {  
  expenseID* integer  
  userID* integer  
  expenseAmount number($float)  
  expenseSource string  
  expenseDescription string  
  expenseDate* string($date-time)  
}}
```

Εικόνα 32: Μοντέλο δεδομένων ExpensesList, που αποτελείται από δεδομένα Expense

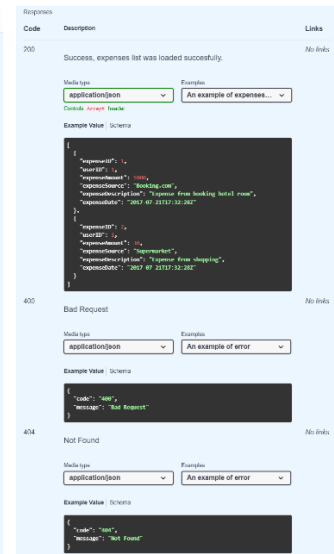
2.2.3.3 Endpoint GET πόρου Expense



Εικόνα 33: Παράμετροι του GET Endpoint του πόρου Expense

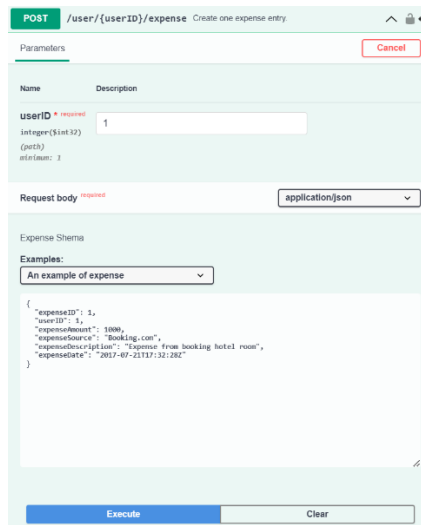


Εικόνα 34: Δοκιμή του GET Endpoint του πόρου Expense

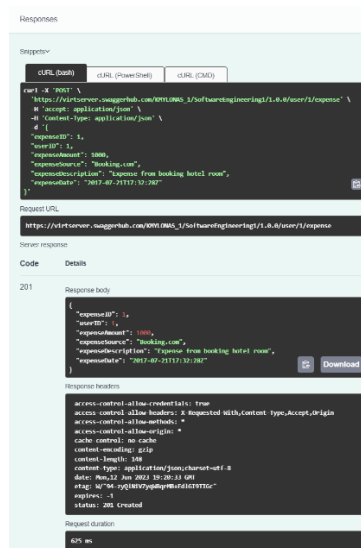


Εικόνα 35: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Expense

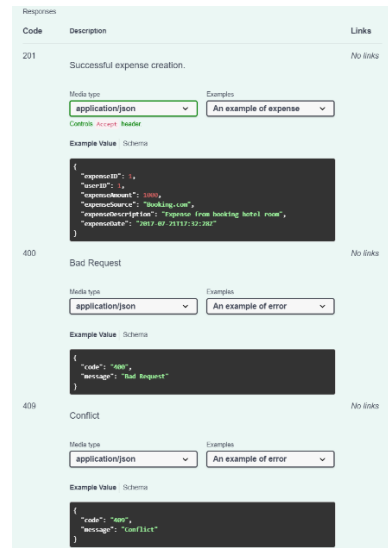
2.2.3.4 Endpoint POST πόρου Expense



Εικόνα 36: Παράμετροι του POST Endpoint του πόρου Expense

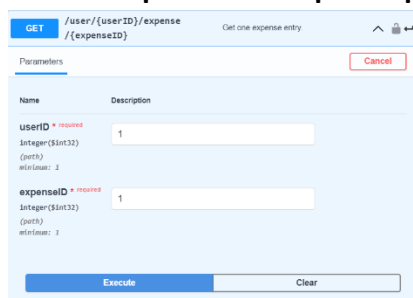


Εικόνα 37: Δοκιμή του POST Endpoint του πόρου Expense

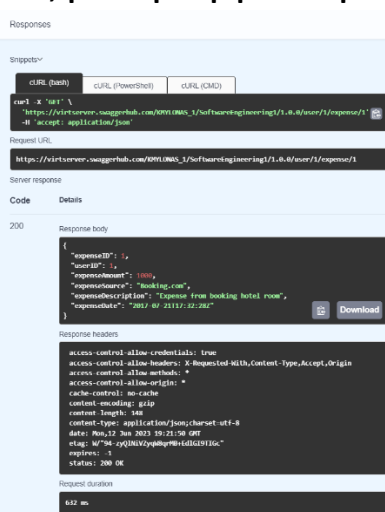


Εικόνα 38: Αποκρίσεις του συστήματος για το POST Endpoint του πόρου Expense

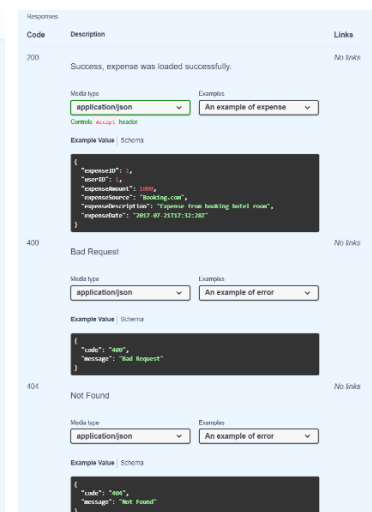
2.2.3.5 Endpoint GET πόρου Expense, για συγκεκριμένο Expense ID



Εικόνα 39: Παράμετροι του GET Endpoint του πόρου Expense, για συγκεκριμένο Expense ID

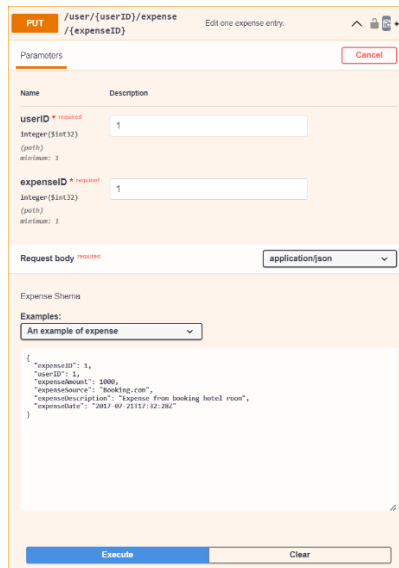


Εικόνα 40: Δοκιμή του GET Endpoint του πόρου Expense, για συγκεκριμένο Expense ID



Εικόνα 41: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Expense, για συγκεκριμένο Expense ID

2.2.3.6 Endpoint PUT πόρου Expense, για συγκεκριμένο Expense ID



PUT /user/{userID}/expense/{expenseID} Edit one expense entry.

Parameters

Name	Description
userID * required	
Integer(\$int32)	1
(min): 1	
(max): 2	
expenseID * required	
Integer(\$int32)	1
(min): 1	
(max): 2	

Request body **required** application/json

Expense Schema

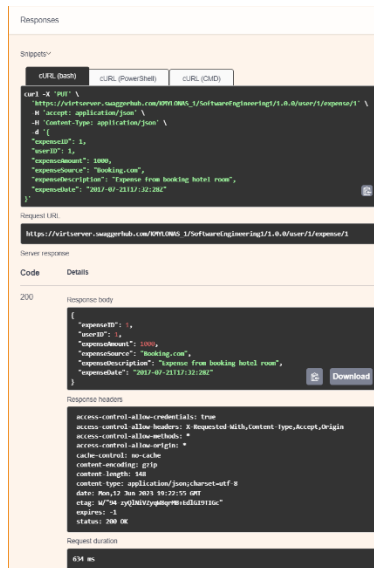
Examples:

An example of expense

```
{
  "expenseID": 1,
  "userID": 1,
  "expenseAmount": 1000,
  "expenseSource": "booking.com",
  "expenseDescription": "Expense from booking hotel room",
  "expenseDate": "2017-07-21T17:32:28Z"
}
```

Execute Clear

Εικόνα 42: Παράμετροι του PUT Endpoint του πόρου Expense, για συγκεκριμένο Expense ID



Responses

Stippets

curl -X PUT \\\n "https://virtserver.swaggerhub.com/EXPENSIR_1/SoftwareEngineering/1.0.0/user/1/expense/1" \\\n -H 'accept: application/json' \\\n -H 'content-type: application/json' \\\n -d '{\n "expenseID": 1,\n "userID": 1,\n "expenseAmount": 1000,\n "expenseSource": "booking.com",\n "expenseDescription": "Expense from booking hotel room",\n "expenseDate": "2017-07-21T17:32:28Z"\n }'\n\nRequest URL\nhttps://virtserver.swaggerhub.com/EXPENSIR_1/SoftwareEngineering/1.0.0/user/1/expense/1\n\nServer response

Code Details

200 Response body

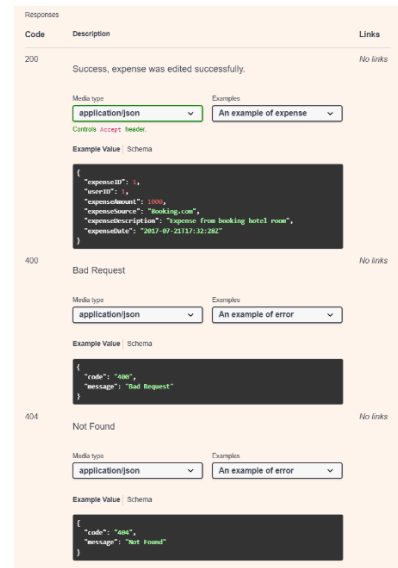
```
{
  "expenseID": 1,
  "userID": 1,
  "expenseAmount": 1000,
  "expenseSource": "booking.com",
  "expenseDescription": "Expense from booking hotel room",
  "expenseDate": "2017-07-21T17:32:28Z"
}
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-headers: *
access-control-allow-methods: *
access-control-allow-origin: *
cache-control: no-cache
content-encoding: gzip
content-length: 340
content-type: application/json; charset=utf-8
date: Mon, 12 Jun 2023 19:22:53 GMT
etag: W/"4-yDmLzGdK1cc39pAZ0000083c"
expires: -1
status: 200 OK
```

Request duration 634 ms

Εικόνα 43: Δοκιμή του PUT Endpoint του πόρου Expense, για συγκεκριμένο Expense ID



Responses

Code Description Links

200 Success, expense was edited successfully. No links

400 Bad Request No links

404 Not Found No links

Media type application/json Examples An example of expense

Example Value Schema

```
{
  "expenseID": 1,
  "userID": 1,
  "expenseAmount": 1000,
  "expenseSource": "booking.com",
  "expenseDescription": "Expense from booking hotel room",
  "expenseDate": "2017-07-21T17:32:28Z"
}
```

Media type application/json Examples An example of error

Example Value Schema

```
{
  "code": "400",
  "message": "Bad Request"
}
```

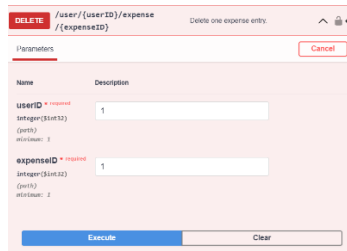
Media type application/json Examples An example of error

Example Value Schema

```
{
  "code": "404",
  "message": "Not Found"
}
```

Εικόνα 44: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου Expense, για συγκεκριμένο Expense ID

2.2.3.7 Endpoint DELETE πόρου Expense



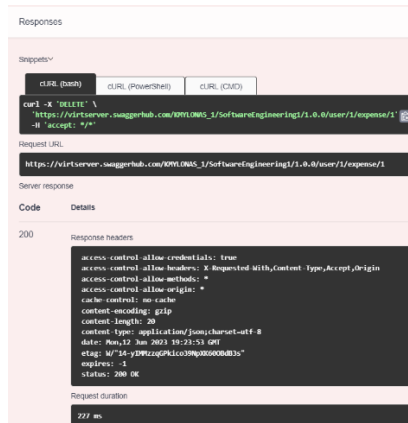
DELETE /user/{userID}/expense/{expenseID} Delete one expense entry.

Parameters

Name	Description
userID * required	
Integer(\$int32)	1
(min): 1	
(max): 2	
expenseID * required	
Integer(\$int32)	1
(min): 1	
(max): 2	

Execute Clear

Εικόνα 45: Παράμετροι του DELETE Endpoint του πόρου Expense, για συγκεκριμένο Expense ID



Responses

Stippets

curl -X DELETE \\\n "https://virtserver.swaggerhub.com/EXPENSIR_1/SoftwareEngineering/1.0.0/user/1/expense/1" \\\n -H 'accept: */*'\n\nRequest URL\nhttps://virtserver.swaggerhub.com/EXPENSIR_1/SoftwareEngineering/1.0.0/user/1/expense/1\n\nServer response

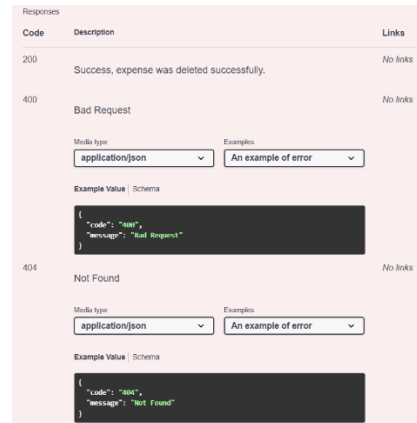
Code Details

200 Response headers

```
access-control-allow-credentials: true
access-control-allow-headers: *
access-control-allow-methods: *
access-control-allow-origin: *
cache-control: no-cache
content-encoding: gzip
content-length: 20
content-type: application/json; charset=utf-8
date: Mon, 12 Jun 2023 19:22:53 GMT
etag: W/"1A-yDmLzGdK1cc39pAZ0000083c"
expires: -1
status: 200 OK
```

Request duration 222 ms

Εικόνα 46: Δοκιμή του DELETE Endpoint του πόρου Expense, για συγκεκριμένο Expense ID



Responses

Code Description Links

200 Success, expense was deleted successfully. No links

400 Bad Request No links

404 Not Found No links

Media type application/json Examples An example of error

Example Value Schema

```
{
  "code": "400",
  "message": "Bad Request"
}
```

Media type application/json Examples An example of error

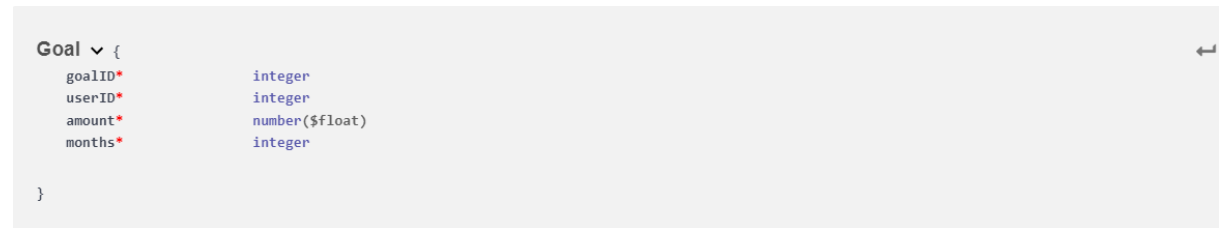
Example Value Schema

```
{
  "code": "404",
  "message": "Not Found"
}
```

Εικόνα 47: Αποκρίσεις του συστήματος για το DELETE Endpoint του πόρου Expense, για συγκεκριμένο Expense ID

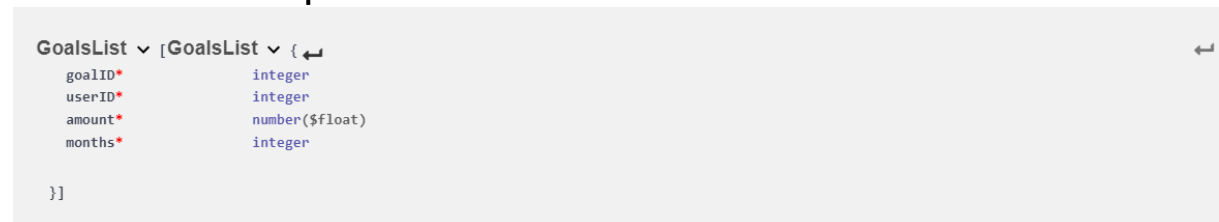
2.2.4 Πόρος Goal

2.2.4.1 Μοντέλο δεδομένων Goal



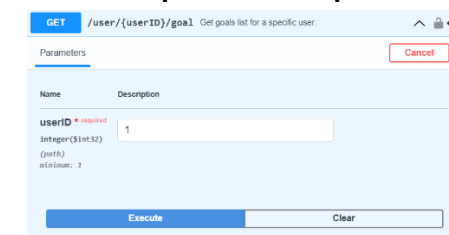
Εικόνα 48: Μοντέλο δεδομένων Goal

2.2.4.2 Μοντέλο δεδομένων GoalsList

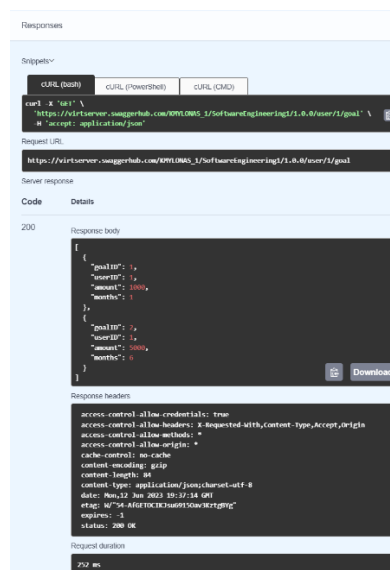


Εικόνα 49: Μοντέλο δεδομένων GoalsList, που αποτελείται από δεδομένα Goal

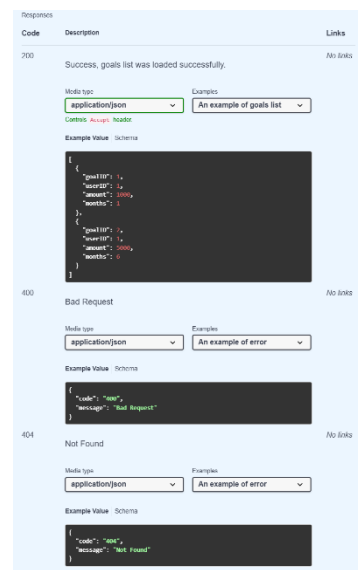
2.2.4.3 Endpoint GET πόρου Goal



Εικόνα 50: Παράμετροι του GET Endpoint του πόρου Goal

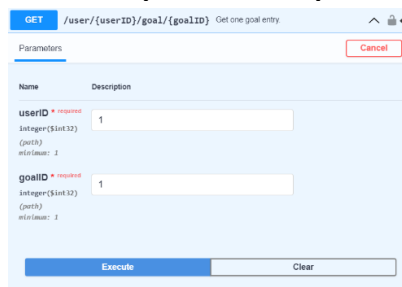


Εικόνα 51: Δοκιμή του GET Endpoint του πόρου Goal

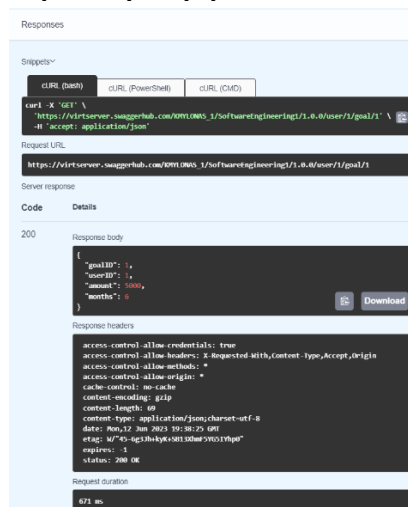


Εικόνα 52: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Goal

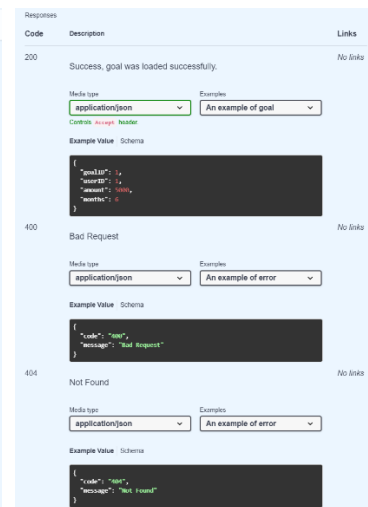
2.2.4.3 Endpoint GET πόρου Goal, για συγκεκριμένο Goal ID



Εικόνα 53: Παράμετροι του GET Endpoint του πόρου Goal, για συγκεκριμένο Goal ID



Εικόνα 54: Δοκιμή του GET Endpoint του πόρου Goal, για συγκεκριμένο Goal ID

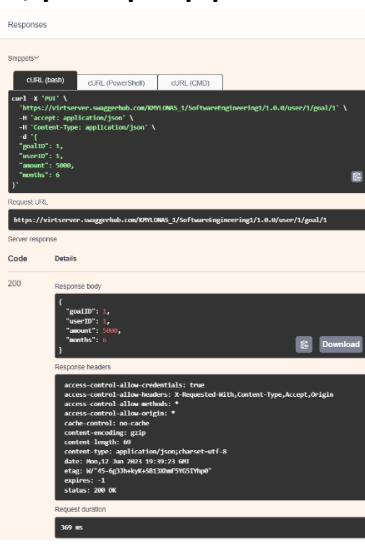


Εικόνα 55: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Goal, για συγκεκριμένο Goal ID

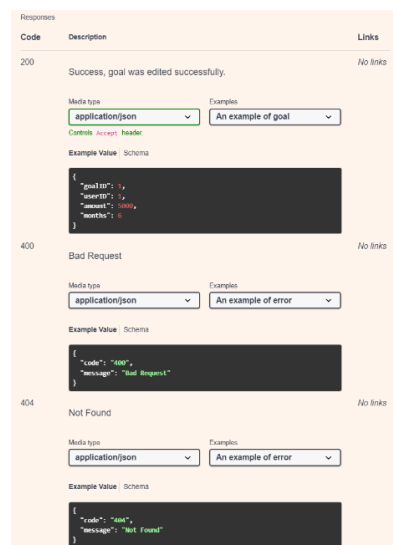
2.2.4.4 Endpoint PUT πόρου Goal, για συγκεκριμένο Goal ID



Εικόνα 56: Παράμετροι του PUT Endpoint του πόρου Goal, για συγκεκριμένο Goal ID



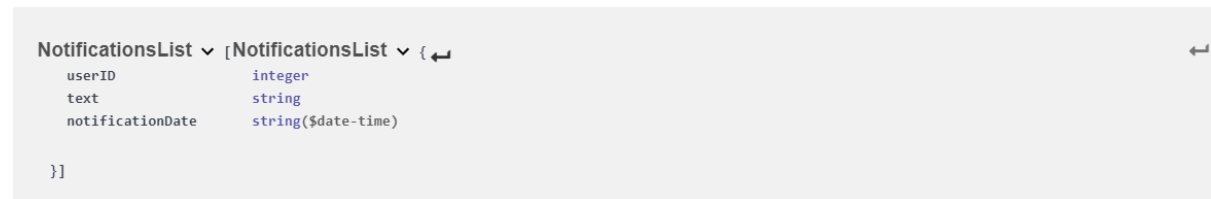
Εικόνα 57: Δοκιμή του PUT Endpoint του πόρου Goal, για συγκεκριμένο Goal ID



Εικόνα 58: Αποκρίσεις του συστήματος για το PUT Endpoint του πόρου Goal, για συγκεκριμένο Goal ID

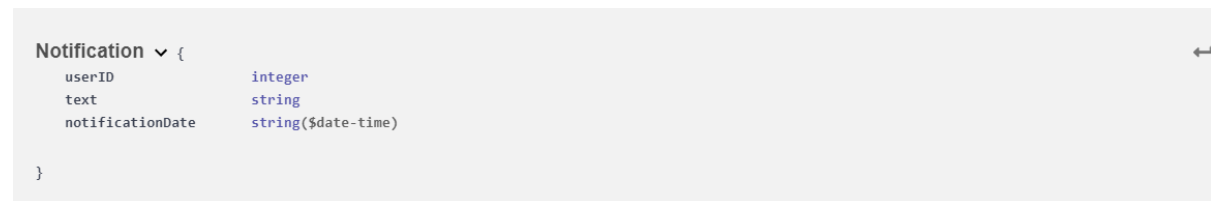
2.2.5 Πόρος Notification

2.2.5.1 Μοντέλο δεδομένων Notification



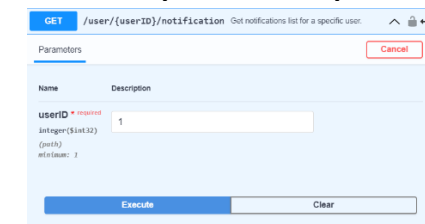
Εικόνα 59: Μοντέλο δεδομένων Notification

2.2.5.2 Μοντέλο δεδομένων Notification

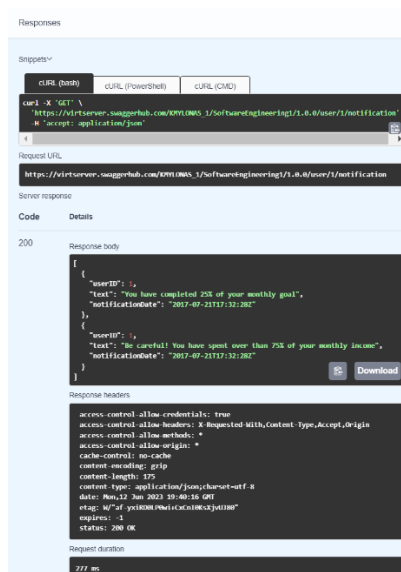


Εικόνα 60: Μοντέλο δεδομένων NotificationsList, που αποτελείται από δεδομένα Notification

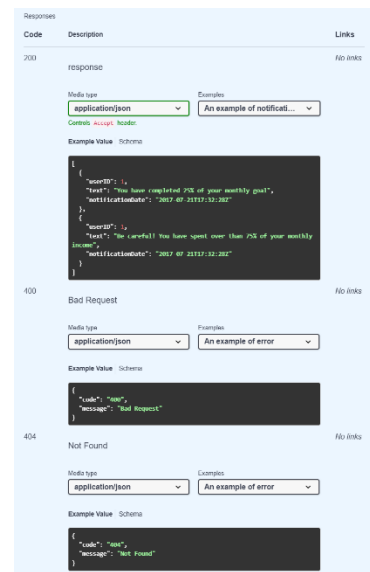
2.2.5.3 Endpoint GET πόρου Notification



Εικόνα 61: Παράμετροι του GET Endpoint του πόρου Notification



Εικόνα 62: Δοκιμή του GET Endpoint του πόρου Notification



Εικόνα 63: Αποκρίσεις του συστήματος για το GET Endpoint του πόρου Notification

3. Υλοποίηση συστήματος με Node-RED

3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές Node-RED

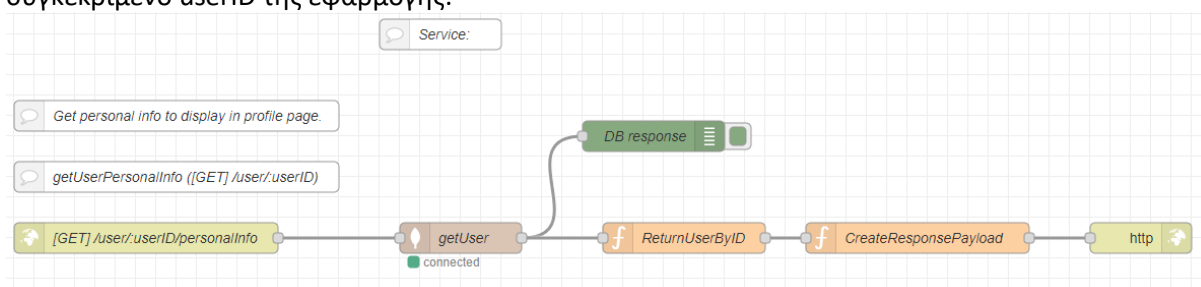
Χρησιμοποιήθηκε η Node Red για την ακόλουθη ενότητα.

- [Σύνδεσμος](#) για το αρχείο τύπου ZIP που περιέχει τα flows.

3.1.1 Ροές πόρου PersonalInfo

3.1.1.1 Ροή endpoint GET /user/{userID}/personalinfo

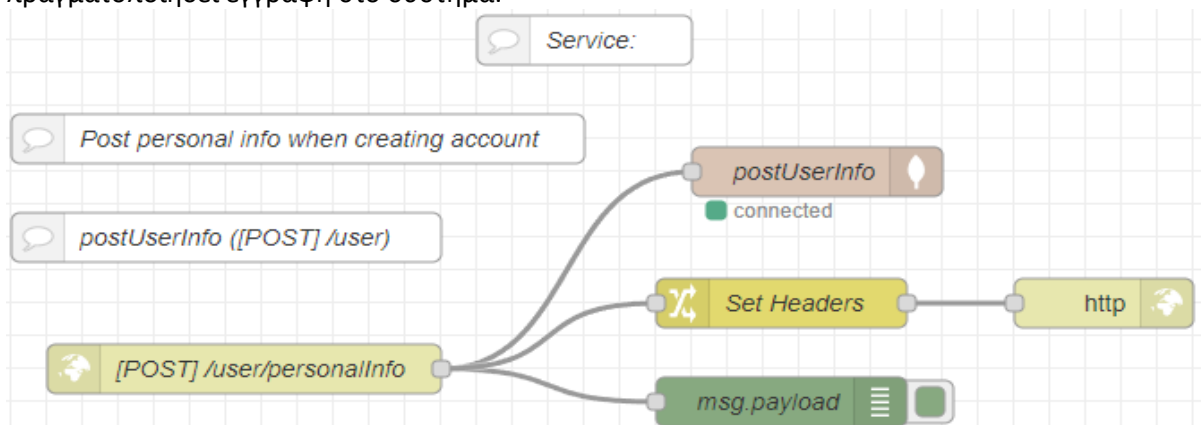
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφονται τα στοιχεία ενός χρήστη με συγκεκριμένο userID της εφαρμογής.



Εικόνα 64: Ροή endpoint GET /user/{userID}/personalinfo

3.1.1.2 Ροή endpoint POST /user/personalinfo

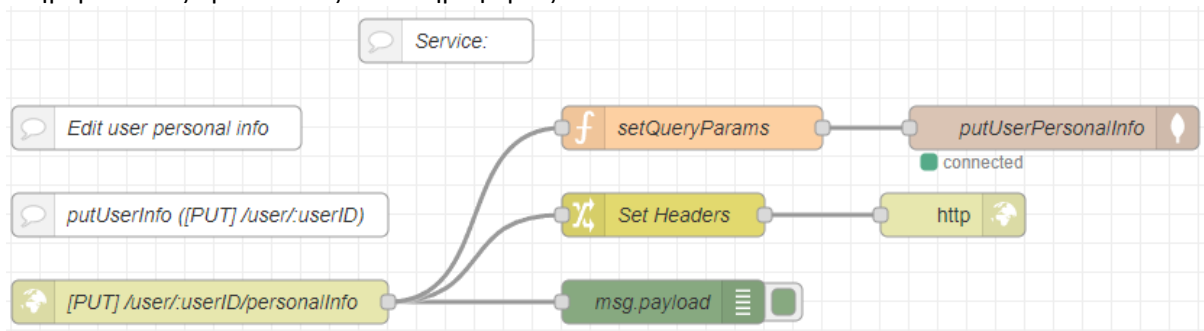
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να πραγματοποιήσει εγγραφή στο σύστημα.



Εικόνα 65: Ροή endpoint POST /user/personalinfo

3.1.1.3 Ροή endpoint PUT /user/{userID}/personalinfo

Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας μπορεί ένας χρήστης με συγκεκριμένο userID να ενημερώσει τις προσωπικές του πληροφορίες.

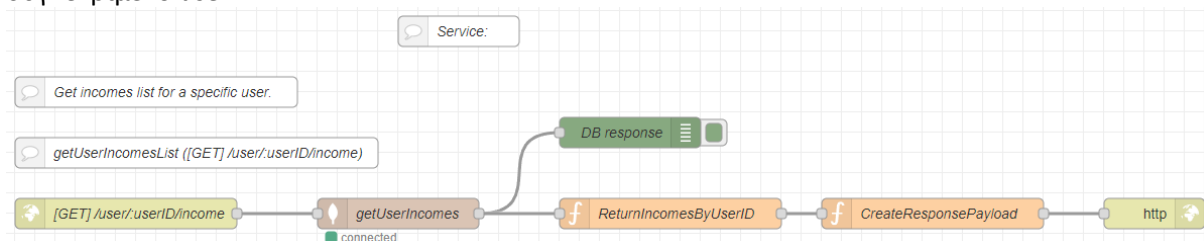


Εικόνα 66: Ροή endpoint PUT /user/{userID}/personalinfo

3.1.2 Ροές πόρου Income

3.1.2.1 Ροή endpoint GET /user/{userID}/income

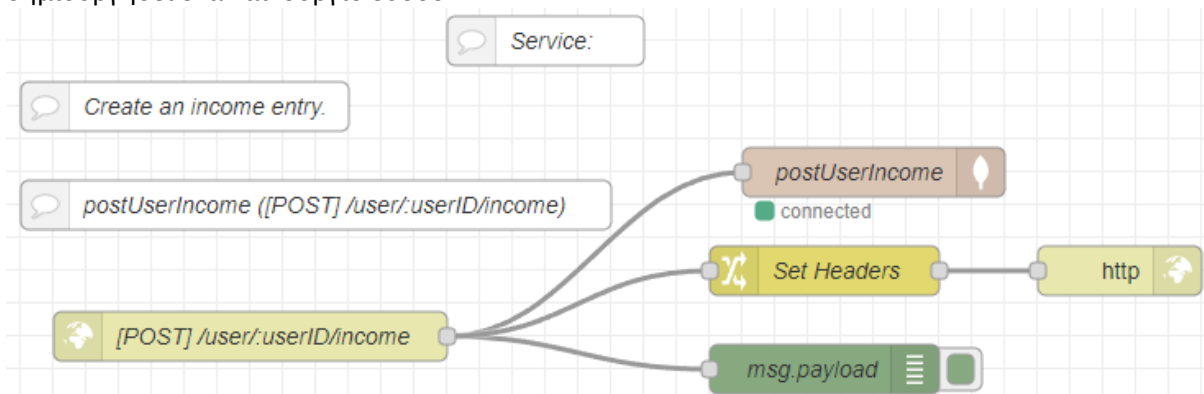
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφονται τα έσοδα για έναν χρήστη με συγκεκριμένο userID.



Εικόνα 67: Ροή endpoint GET /user/{userID}/income

3.1.2.2 Ροή endpoint POST /user/{userID}/income

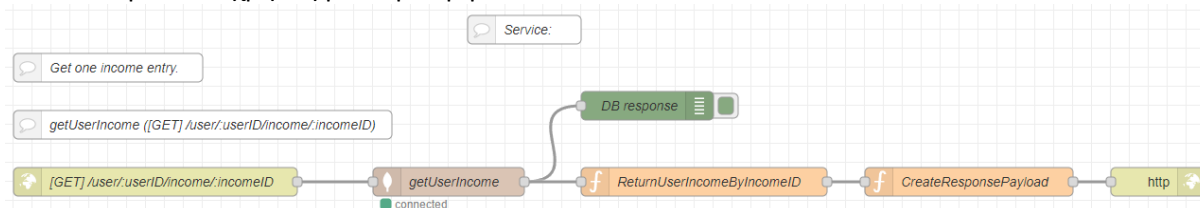
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να δημιουργήσει ένα καινούργιο έσοδο.



Εικόνα 68: Ροή endpoint POST /user/{userID}/income

3.1.2.3 Ροή endpoint GET /user/{userID}/income/{incomeID}

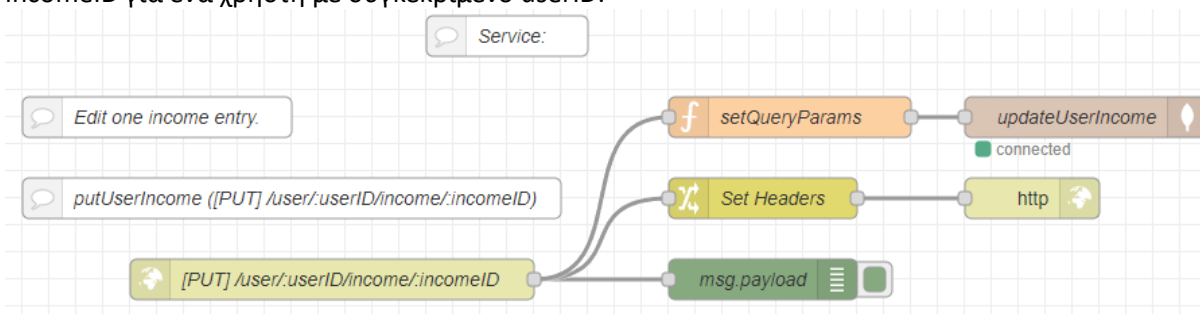
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφεται ένα έσοδο με συγκεκριμένο incomeID για έναν χρήστη με συγκεκριμένο userID.



Εικόνα 69: Ροή endpoint GET /user/{userID}/income/{incomeID}

3.1.2.4 Ροή endpoint PUT /user/{userID}/income/{incomeID}

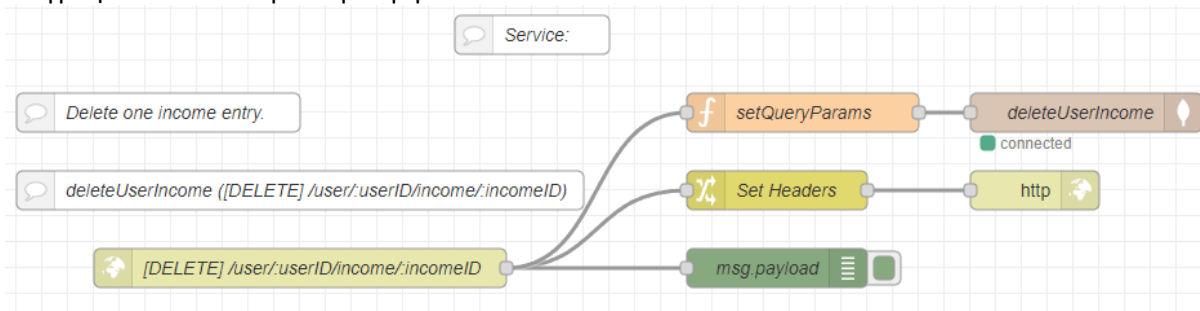
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφεται ένα έσοδο με συγκεκριμένο incomeID για ένα χρήστη με συγκεκριμένο userID.



Εικόνα 70: Ροή endpoint PUT /user/{userID}/income/{incomeID}

3.1.2.5 Ροή endpoint DELETE /user/{userID}/income/{incomeID}

Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να διαγράψει ένα έσοδο με συγκεκριμένο incomeID.

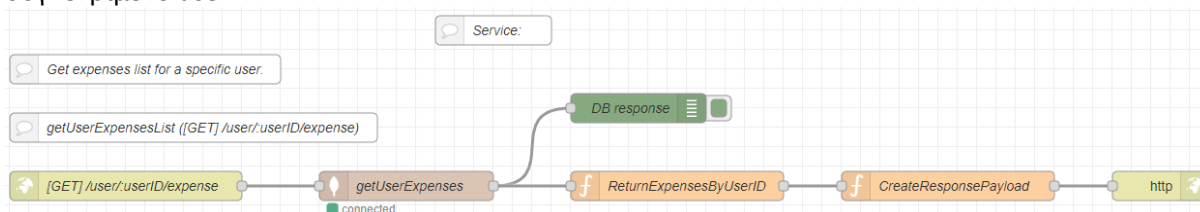


Εικόνα 71: Ροή endpoint DELETE /user/{userID}/income/{incomeID}

3.1.3 Ροές πόρου Expense

3.1.2.1 Ροή endpoint GET /user/{userID}/expense

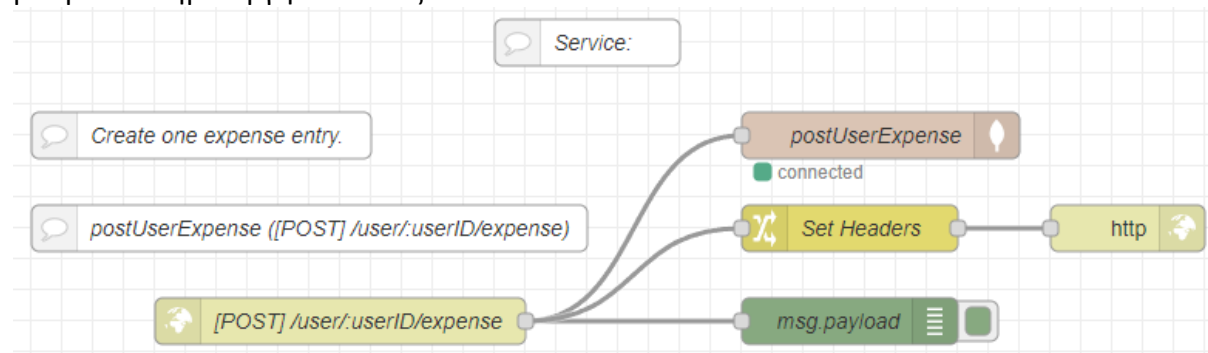
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφονται όλα τα έξοδα για έναν χρήστη με συγκεκριμένο userID.



Εικόνα 72: Ροή endpoint GET /user/{userID}/expense

3.1.2.2 Ποή endpoint POST /user/{userID}/expense

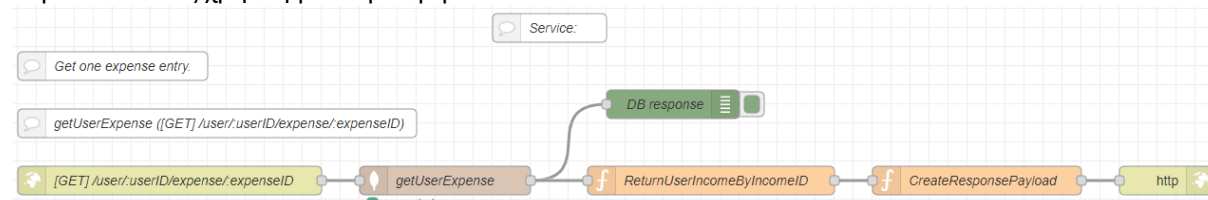
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να δημιουργήσει ένα έξοδο.



Εικόνα 73: Ποή endpoint POST /user/{userID}/expense

3.1.2.3 Ποή endpoint GET /user/{userID}/expense/{expenseID}

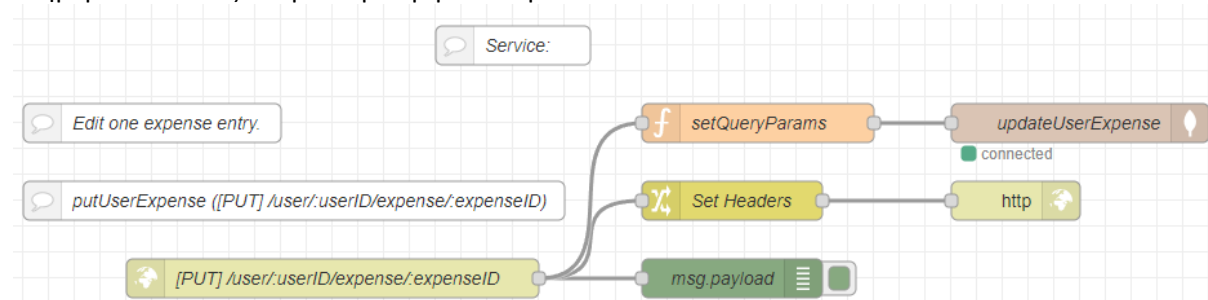
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφεται ένα έξοδο με συγκεκριμένο expenseID ενός χρήστη με συγκεκριμένο userID.



Εικόνα 74: Ποή endpoint GET /user/{userID}/expense/{expenseID}

3.1.2.4 Ποή endpoint PUT /user/{userID}/expense/{expenseID}

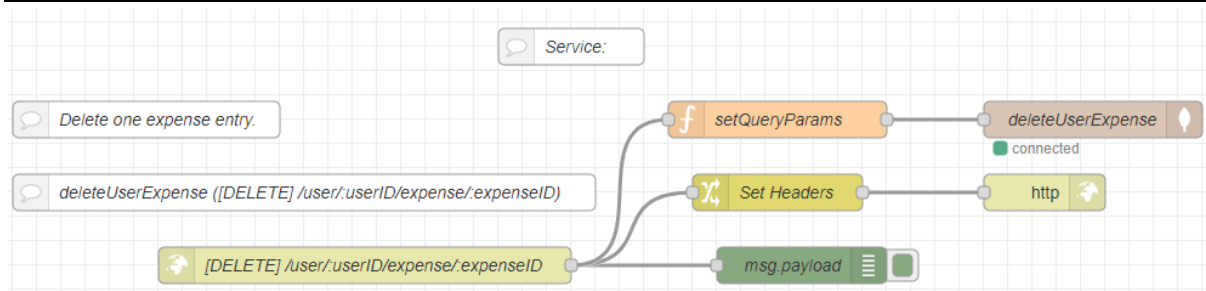
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να ενημερώσει ένα έξοδο με συγκεκριμένο expenseID.



Εικόνα 75: Ποή endpoint PUT /user/{userID}/expense/{expenseID}

3.1.2.5 Ποή endpoint DELETE /user/{userID}/expense/{expenseID}

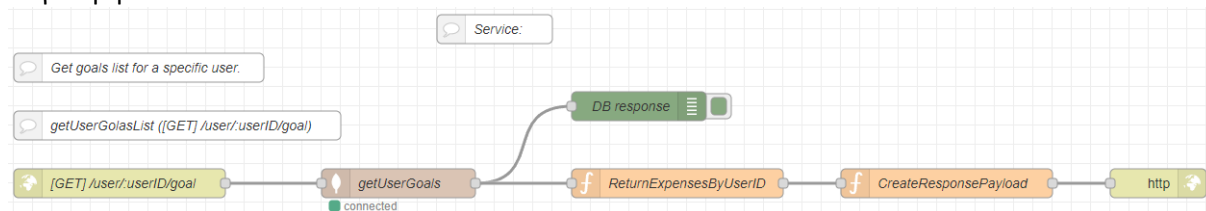
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να διαγράψει ένα έξοδο με συγκεκριμένο expenseID.

Εικόνα 76: Ροή endpoint DELETE `/user/{userID}/expense/{expenseID}`

3.1.4 Ροές πόρου Goal

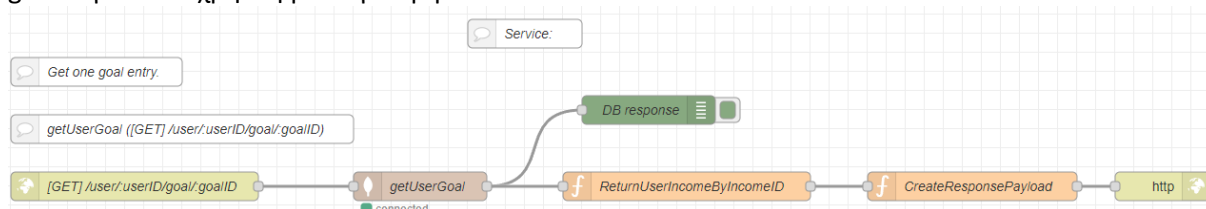
3.1.4.1 Ροή endpoint GET `/user/{userID}/goal`

Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφονται οι στόχοι ενός χρήστη με συγκεκριμένο userID.

Εικόνα 77: Ροή endpoint GET `/user/{userID}/goal`

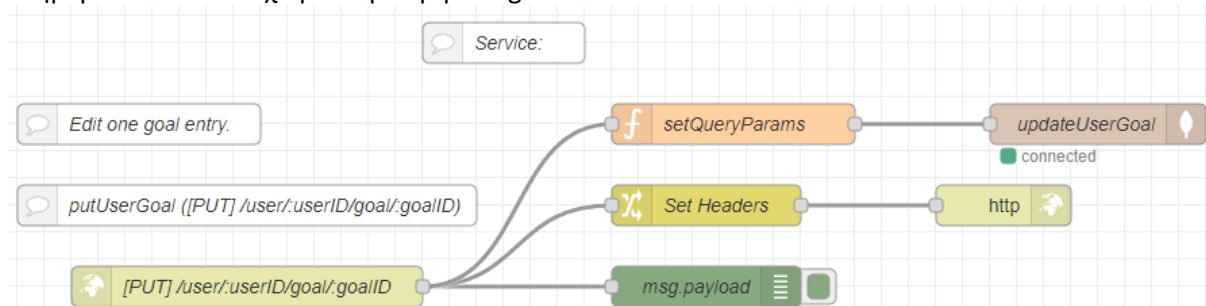
3.1.4.2 Ροή endpoint GET `/user/{userID}/goal/{goalID}`

Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφεται ένας στόχος με συγκεκριμένο goalID για έναν χρήστη με συγκεκριμένο userID.

Εικόνα 78: Ροή endpoint GET `/user/{userID}/goal/{goalID}`

3.1.4.3 Ροή endpoint PUT `/user/{userID}/goal/{goalID}`

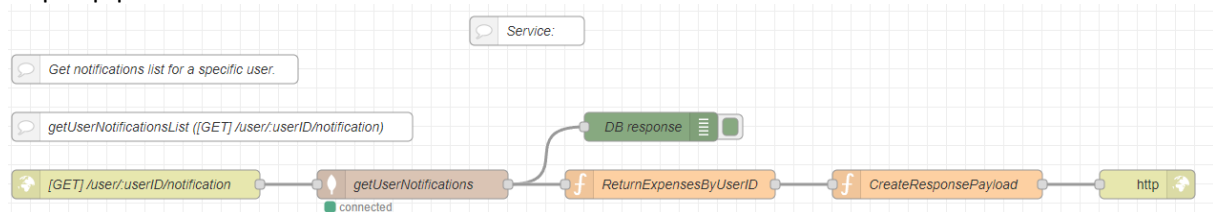
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης με συγκεκριμένο userID μπορεί να ενημερώσει έναν στόχο με συγκεκριμένο goalID.

Εικόνα 79: Ροή endpoint PUT `/user/{userID}/goal/{goalID}`

3.1.5 Ροές πόρου Notification

3.1.5.1 Ροή endpoint GET /user/{userID}/notifications

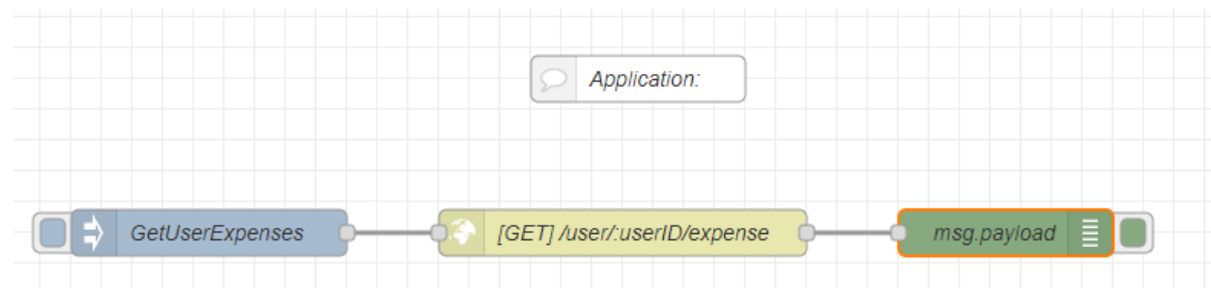
Η ροή που υλοποιεί την υπηρεσία μέσω της οποίας επιστρέφονται οι ειδοποιήσεις ενός χρήστη με συγκεκριμένο userID.



Εικόνα 80: Ροή endpoint GET /user/{userID}/notifications

3.2 Υλοποίηση ιστοριών χρήστη

3.2.1 Ιστορία Χρήστη Add Expenses



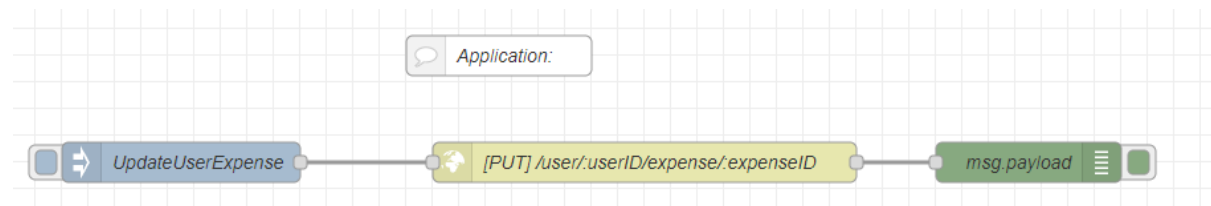
Εικόνα 81: Ιστορία Χρήστη Add Expenses σε Node Red

Ροή μέσω της οποίας ο χρήστης μπορεί να δημιουργήσει ένα καινούργιο έξοδο.

Πίνακας 2: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.1

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
PostUserExpense	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[GET] /user/:userID/expense	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/expense η οποία δίνει τη δυνατότητα στον χρήστη να δημιουργήσει ένα έξοδο.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.2 Ιστορία Χρήστη Edit Expenses



Εικόνα 82: Ιστορία Χρήστη Edit Expenses σε Node Red

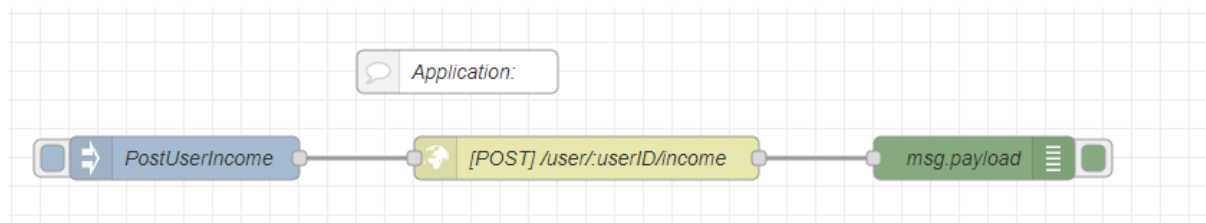
Ροή μέσω της οποίας ο χρήστης μπορεί να επεξεργαστεί ένα υπάρχον έξοδο.

Πίνακας 3: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.2

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateUserExpense	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[PUT] /user/:userID/expense/:expenseID	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/expense/{expenseID} η οποία δίνει τη δυνατότητα στον χρήστη να επεξεργαστεί ένα έξοδο.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση

του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.3 Ιστορία Χρήστη Add Incomes



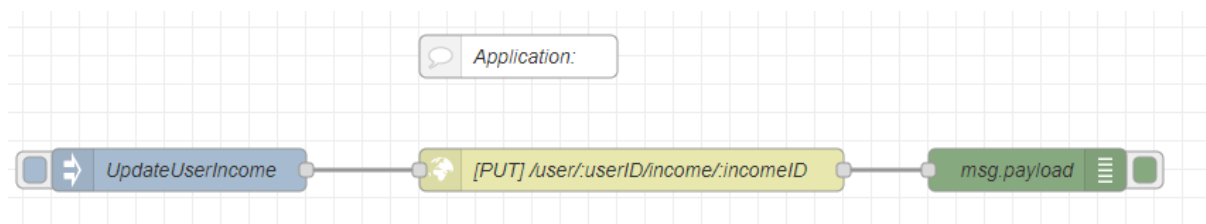
Εικόνα 83: Ιστορία Χρήστη Add Incomes σε Node Red

Ροή μέσω της οποίας ο χρήστης μπορεί να δημιουργήσει ένα καινούργιο έσοδο.

Πίνακας 4: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.3

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
PostUserIncome	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[POST] /user/:userID/income	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/income η οποία δίνει τη δυνατότητα στον χρήστη να δημιουργήσει ένα έσοδο.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.4 Ιστορία Χρήστη Edit Incomes



Εικόνα 84: Ιστορία Χρήστη Edit Incomes σε Node Red

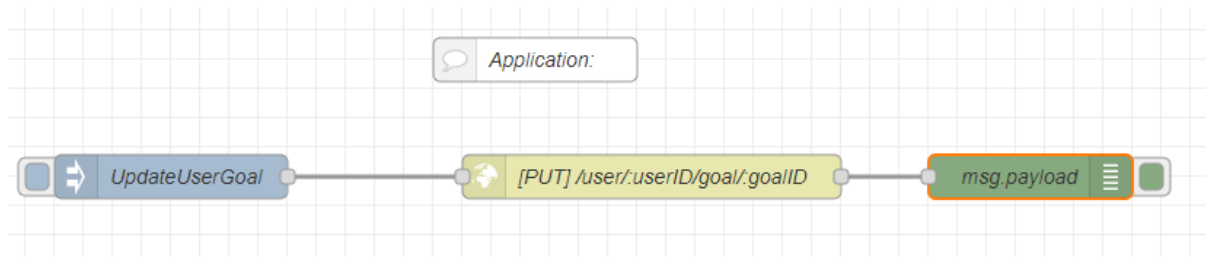
Ροή μέσω της οποίας ο χρήστης μπορεί να επεξεργαστεί ένα υπάρχον έσοδο.

Πίνακας 5: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.4

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateUserIncome	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[PUT] /user/{userID}/income/{incomeID}	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/income/{incomeID} η οποία δίνει τη δυνατότητα στον χρήστη να επεξεργαστεί ένα έσοδο.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση

του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.5 Ιστορία Χρήστη Manage Goals



Εικόνα 85: Ιστορία Χρήστη Manage Goals σε Node Red

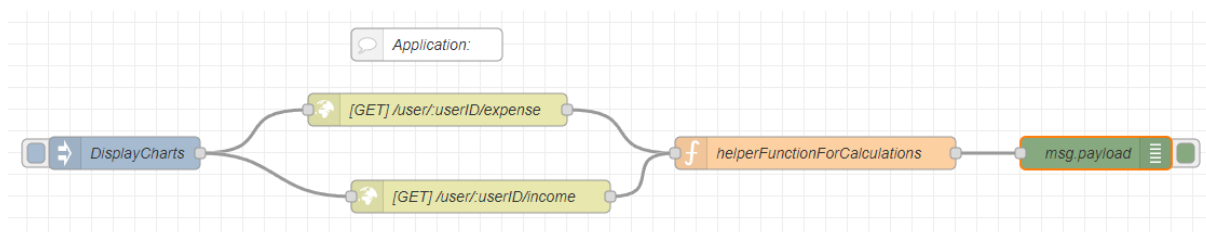
Ροή μέσω της οποίας ο χρήστης μπορεί να επεξεργαστεί έναν στόχο.

Πίνακας 6: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.5

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateUserGoal	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[PUT] /user/{userID}/goal/{goalID}	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/goal/{goalID} η οποία δίνει τη δυνατότητα στον χρήστη να επεξεργαστεί έναν στόχο του.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.6 Ιστορία Χρήστη Display Charts

Εικόνα 86: Ιστορία Χρήστη Display Charts σε Node Red



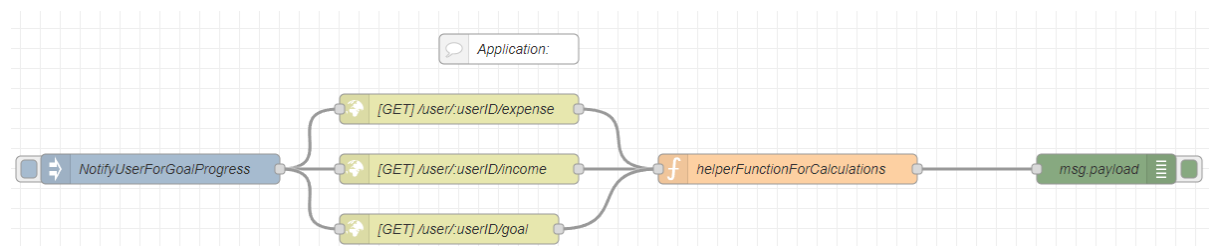
Ροή μέσω της οποίας το σύστημα εμφανίζει τα διαγράμματα προόδου του χρήστη.

Πίνακας 7: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.6

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
DisplayCharts	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[GET] /user/{userID}/expense	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/expense η οποία επιστρέφει τη λίστα με τα έξοδα του χρήστη.

[GET] /user/:userID/income	http request	Κάνει κλήση προς την υπηρεσία /user/{userID}/income η οποία επιστρέφει τη λίστα με τα έσοδα του χρήστη.
helperFunctionForCalculations	function	Επεξεργάζεται τα δεδομένα από τα responses και δίνει στην έξοδο το αποτέλεσμα για το κάθε ιστόγραμμα.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.7 Ιστορία Χρήστη Notify User for progress



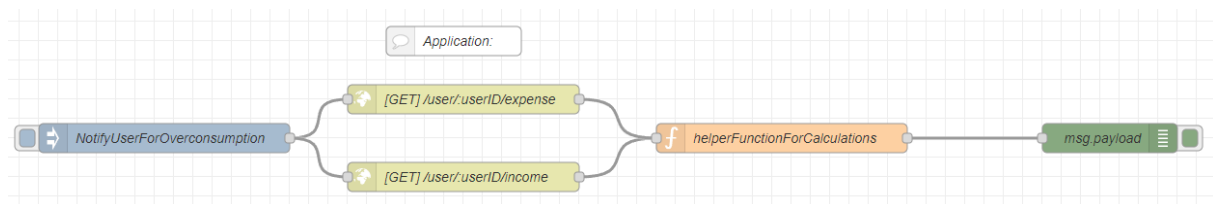
Εικόνα 87: Ιστορία Χρήστη Notify User for progress σε Node Red

Ροή μέσω της οποίας το σύστημα ειδοποιεί τον χρήστη για τη πρόοδό του.

Πίνακας 8: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.7

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
NotifyUserForGoalProgress	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[GET] /user/:userID/expense	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/expense η οποία επιστρέφει τη λίστα με τα έξοδα του χρήστη.
[GET] /user/:userID/income	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/income η οποία επιστρέφει τη λίστα με τα έσοδα του χρήστη.
[GET] /user/:userID/goal	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/goal η οποία επιστρέφει τη λίστα με τους στόχους του χρήστη.
helperFunctionForCalculations	function	Επεξεργάζεται τα δεδομένα από τα responses και δίνει στην έξοδο το αποτέλεσμα για κάθε πρόοδο στόχου.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.

3.2.8 Ιστορία Χρήστη Notify User for overconsumption



Εικόνα 88: Ιστορία Χρήστη Notify User for overconsumption σε Node Red

Ροή μέσω της οποίας το σύστημα ειδοποιεί τον χρήστη για υπερκατανάλωση.

Πίνακας 9: Πίνακας με όνομα, τύπο και περιγραφή κόμβου για τη ροή 3.2.8

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
NotifyUserForOverconsumption	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
[GET] /user/:userID/expense	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/expense η οποία επιστρέφει τη λίστα με τα έξοδα του χρήστη.
[GET] /user/:userID/income	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/income η οποία επιστρέφει τη λίστα με τα έσοδα του χρήστη.
helperFunctionForCalculations	function	Επεξεργάζεται τα δεδομένα από τα responses και δίνει στην έξοδο το αποτέλεσμα για το όριο δαπάνης του χρήστη.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος με βάση τη κλήση του χρήστη, δηλαδή το ζήτημα που απαιτεί ο χρήστης.



Παράρτημα Ι – Γλωσσάριο

ΛΑ-xxxx: Λειτουργική Απαίτηση xxxx
ΜΛΑ-xxxx: Μη-Λειτουργική Απαίτηση xxxx
ΟΑ: Ομάδα Εργασίας